Linear solvers in fluidity

All you ever wanted to know about iterative solvers, preconditioners and multi grid.

Stephan Kramer

Chris Pain, Matthew Piggott

Applied Modelling and Computation Group (AMCG),

Faculty of Earth Science and Engineering,

Imperial College London.

Imperial College London

Outline

- Introduction on iterative methods, preconditionning and multigrid
- The efficient solution of large aspect ratio pressure Poisson equations using multi-grid techniques
- Linear solvers in fluidity

Introduction on iterative methods, preconditionning and multigrid

Imperial College London

Linear solvers

Solution of a linear system

$$A\mathbf{x} = \mathbf{b} \tag{1}$$

Direct methods

Construct inverse A^{-1} of matrix and so computes $\mathbf{x} = A^{-1}\mathbf{b}$. Construction of dense inverse matrix is expensive in both memory and time.

Iterative methods

Construct approximation $\mathbf{x}^k$ of $\mathbf{x}$ with improvement each step, so that (hopefully) $\mathbf{x}^k$ converges to $\mathbf{x}$.

Residual

Solution of a linear system

$$A\mathbf{x} = \mathbf{b}$$

Approximation x^k in k-th iteration.

Residual: $\mathbf{r}^k = A\mathbf{x}^k - \mathbf{b}$

Error: $\mathbf{e}^k = \mathbf{x}^k - \mathbf{x}$

Note: $A\mathbf{e}^k = A\mathbf{x}^k - A\mathbf{x} = \mathbf{r}^k$

Stationary iterative method

Suppose M is an approximation of the matrix A such that M^{-1} is easy to calculate.

Then the error can be approximated by

$$\mathbf{e}^k = \mathbf{x}^k - \mathbf{x} \approx M^{-1} A \mathbf{e}^k = M^{-1} \mathbf{r}^k$$

Stationary iterative method:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - M^{-1} \mathbf{r}^k$$

Jacobi and Gauss Seidel

Jacobi iteration

Take approximate matrix M to be only the diagonal of A.

Gauss Seidel iteration

Take M to be everything on or below the diagonal:

$$\begin{pmatrix} A_{11} & 0 & 0 & \dots \\ A_{21} & A_{22} & 0 & \dots \\ A_{31} & A_{32} & A_{33} & \dots \\ \dots & \dots & \dots & \ddots \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ \dots \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \dots \end{pmatrix}$$

thus computing error approximation $\mathbf{z}^k = M^{-1}\mathbf{r}^k$.

Krylov subspace methods

Consider iteration:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{r}^k$$

$$\implies \mathbf{r}^{k+1} = \mathbf{r}^k + \alpha_k A\mathbf{r}^k$$

Working out:

$$\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$$

$$\mathbf{r}^1 = \mathbf{r}^0 + \alpha_0 A\mathbf{r}^0$$

$$\mathbf{r}^2 = \mathbf{r}^1 + \alpha_1 A\mathbf{r}^1 = \mathbf{r}^0 + (\alpha_0 + \alpha_1)A\mathbf{r}^0 + \alpha_1 A^2\mathbf{r}^0$$

$$\dots$$

Thus $\mathbf{r}^k$ is linear combination of $\mathbf{r}^0, A\mathbf{r}^0, A^2\mathbf{r}^0, \dots A^k\mathbf{r}^0$.

Krylov subspace methods

Krylov subspace is linear space spanned by these vectors:

$$K^k = span\left(\mathbf{r}^0, A\mathbf{r}^0, A^2\mathbf{r}^0, \ldots A^k\mathbf{r}^0\right)$$

Krylov subspace methods try to find optimal approximation in this space
Well known methods:

- Conjugate Gradient (CG) for symmetric matrices
- GMRES
- others: BiCGSTAB, CGSquared, ...

Condition number

Consider eigenvalue decomposition of A:

$$A\mathbf{v}_i = \lambda_i \mathbf{v}_i, \text{ with } i = 1, 2, \ldots, n$$

and decompose the error and residual using those eigenvectors:

$$\mathbf{e}^k = \sum_i \epsilon_i \mathbf{v}_i$$

$$\mathbf{r}^k = A\mathbf{e}^k = \sum_i \lambda_i \epsilon_i \mathbf{v}_i$$

the components of the error with large eigenvalue will be enlarged, and those with small eigenvalue diminished.

Condition number: $\dfrac{\max_i |\lambda_i|}{\min_i |\lambda_i|}$

Preconditioning

Combine stationary approach with approximate inverse matrix M^{-1} with Krylov methods

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k M^{-1} \mathbf{r}^k$$

this is equivalent with applying original Krylov method to solve

$$M^{-1} A \mathbf{x} = M^{-1} \mathbf{b}$$

Thus now we should consider the condition number of $M^{-1}A$. The approximate inverse matrix, also called preconditioner, is useful if it brings down the condition number of $M^{-1}A$.

Preconditioned Krylov Subspace

Combines Krylov subspace methods:

- Conjugate Gradient (CG) for symmetric matrices
- GMRES
- others: BiCGSTAB, CGSq, ...

with suitable preconditioner:

- Jacobi
- Gauss Seidel
- Succesive Symmetric Over-Relaxation (SSOR)
- Incomplete LU (ILU)
- Multigrid methods
- many others

Preconditioners for Poisson equation

Poisson equation

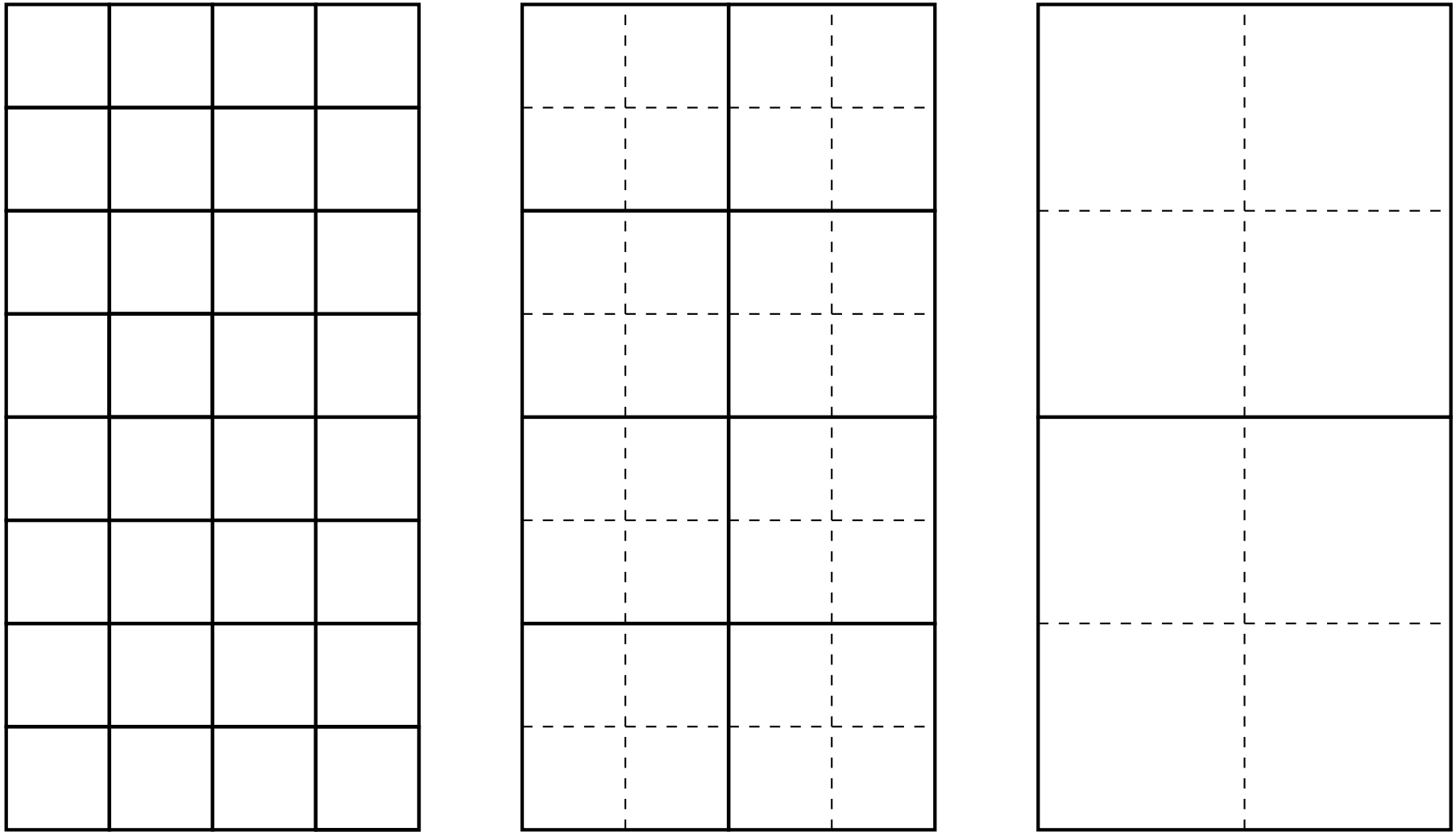
$$\nabla^2 \Phi = f$$

In the Poisson equation the eigenvectors of A corrrespond with the discrete Fourier modes:

- Short range modes with large eigenvalues $\mathcal{O}(\frac{1}{\Delta x^2})$.
- Long range modes with small eigenvalues $\mathcal{O}(\frac{1}{L^2})$.

Traditional preconditioners only act on short range modes in the error (act as smoothers).
Multigrid solvers reduce the different components of the error on a hierarchy of fine to coarse grids.

Multigrid hierarchy

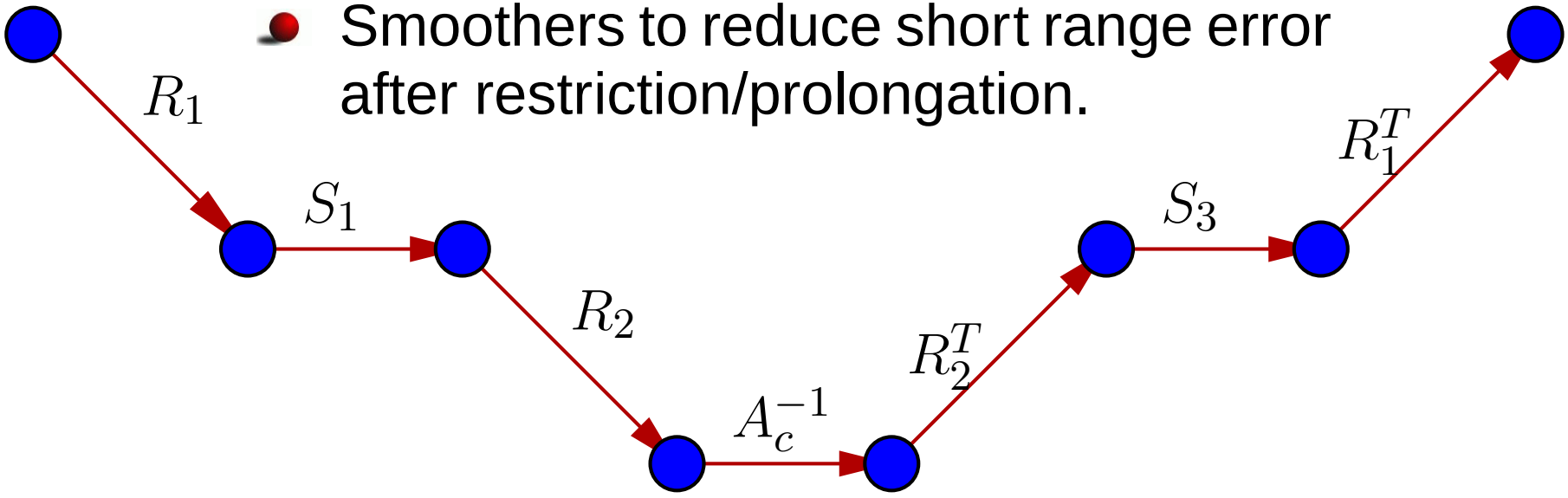


Multigrid solvers

- Selection of coarse grid with restriction (R) and prolongation operators (R^T). Classical multigrid uses grid coarsening (typically $h \rightarrow 2h$), Algebraic Multi-Grid (AMG) bases coarsening on matrix graph.
- Construction of coarse grid approximation of A:

$$A_c = RAR^T$$

- Smoothers to reduce short range error after restriction/prolongation.



The efficient solution of large aspect ratio pressure Poisson equations using multi-grid techniques

Large aspect ratio

Two approaches for pressure solution:

- Hydrostatic models reduce the 3D pressure Poisson equation to a 2D horizontal equation by simplifying the vertical balance.
- Non-hydrostatic models, in which the full 3D pressure Poisson equation is solved, are usually applied to small scale, i.e. local in the horizontal, problems. In large aspect ratio geometries non-hydrostatic models becomes extremely inefficient.

Objective

Challenge is to combine both approaches, in such a way that we maintain the accuracy of non-hydrostatic models and are able to compete with shallow water models for (nearly) hydrostatic regions.

For layered models:

John Marshall, Alistair Adcroft, Chris Hill, Lev Perelman

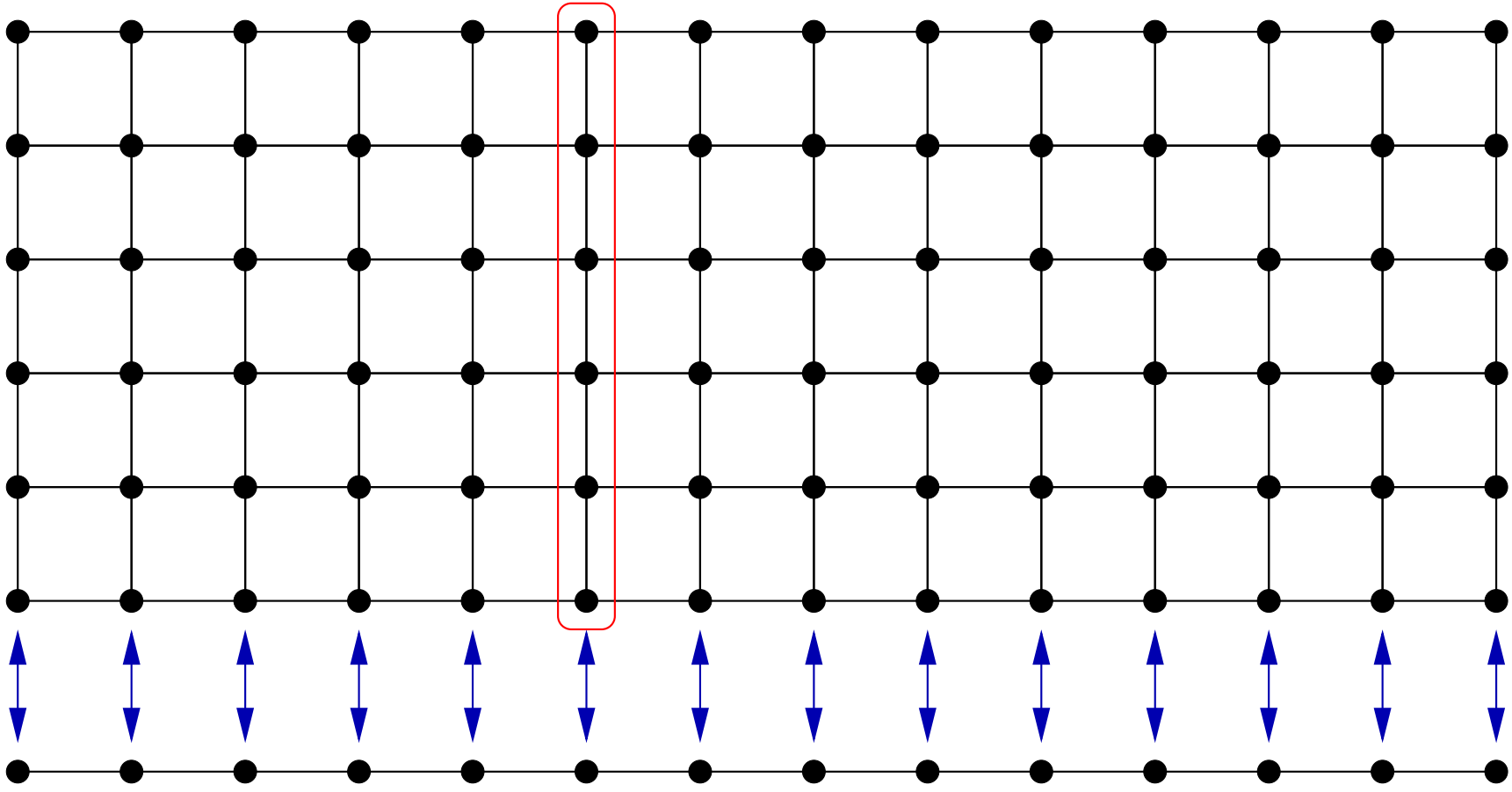
A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers J. Geophys. Res., vol. 102, C3, 1997

Aim for a solution strategy for 3D fully unstructured meshes that performs well across the whole range of scales.

Approach for layered meshes

Similar to John Marshall et al. 1997:

- Solve vertically lumped 2D system.
- Extrapolate solution over vertical.
- Apply block Gauss Seidel.
- Some iterations for full 3D system.



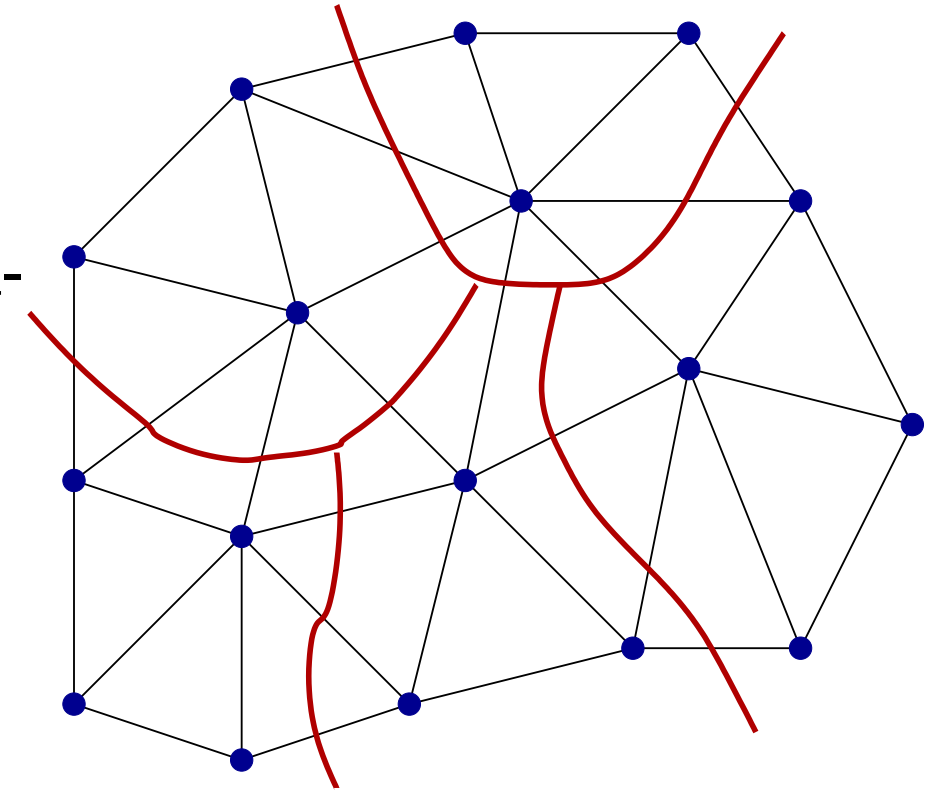
Algebraic Multigrid

Smoothed aggregation approach for 3D Poisson problem with unstructured meshes. Based on:

P. Vaněk, J. Mandel and M. Brezina,

Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems Computing 56 (1996)

- Clusters of fine nodes form coarse nodes.
- Smoothing in prolongation/restriction.



For our purposes:

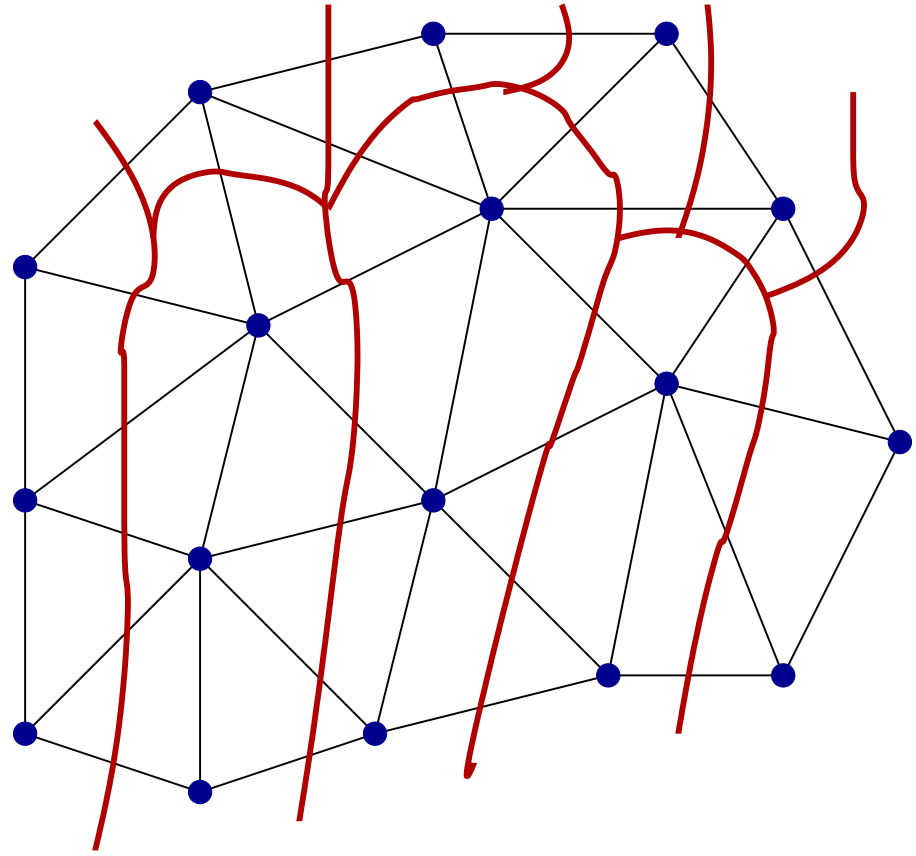
- Optional larger clusters.
- BSSOR as smoother.

Coarsening in the right direction

For large aspect ratio, (and in general anisotropic or multi scale problems) it is important to guide coarsening not only by the topology of the matrix graph but also by coefficient size.

For large aspect ratio this will lead to coarsening in the vertical.

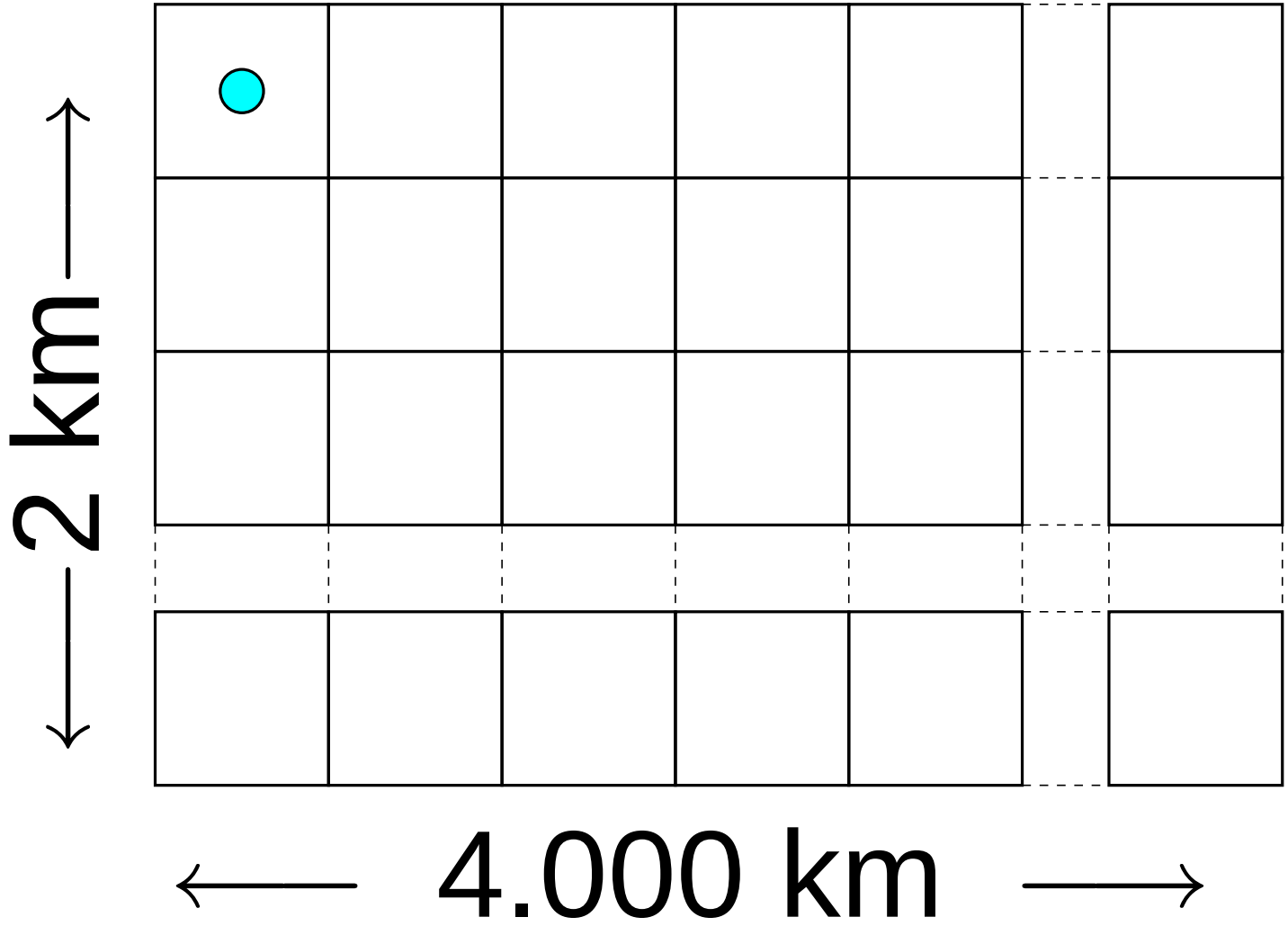
More generally it provides a smooth transition across the various length scales.



Large aspect ratio test case

2DV Poisson problem:

- 200×20 grid.
- 2000×200 grid.



Large aspect ratio test case

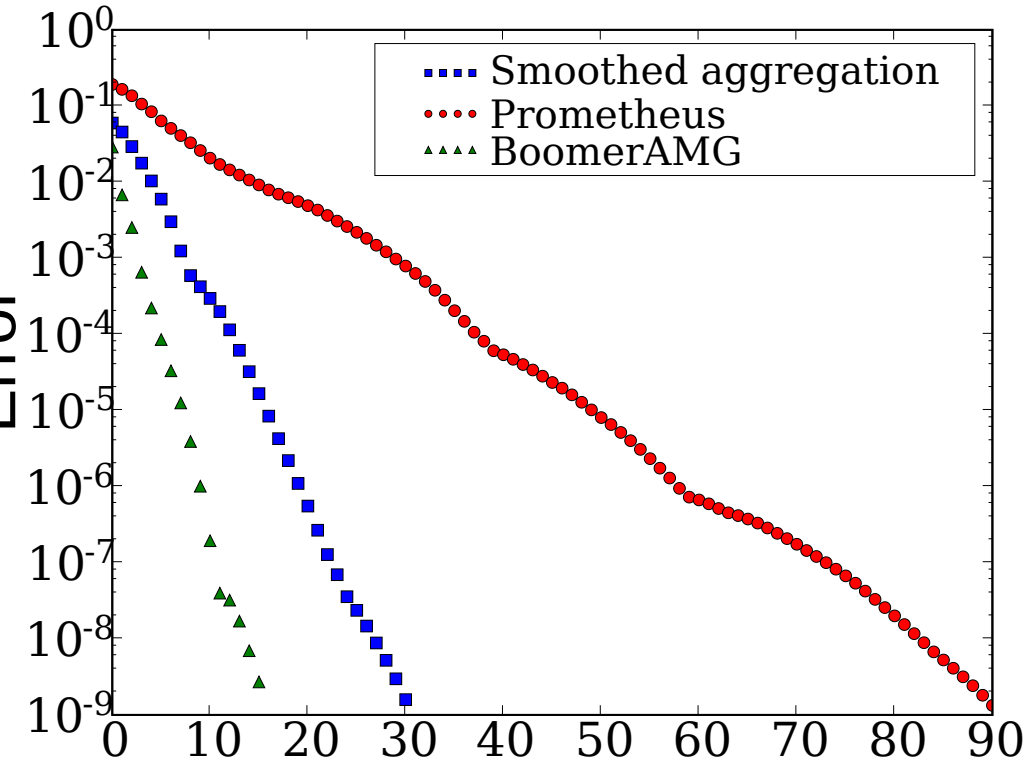
Preconditioner	iterations	time
Sm. aggregation	4	0.03
BoomerAMG	4	0.02
Prometheus	190	0.55
ICC	200	0.26
SSOR	2900	3.80

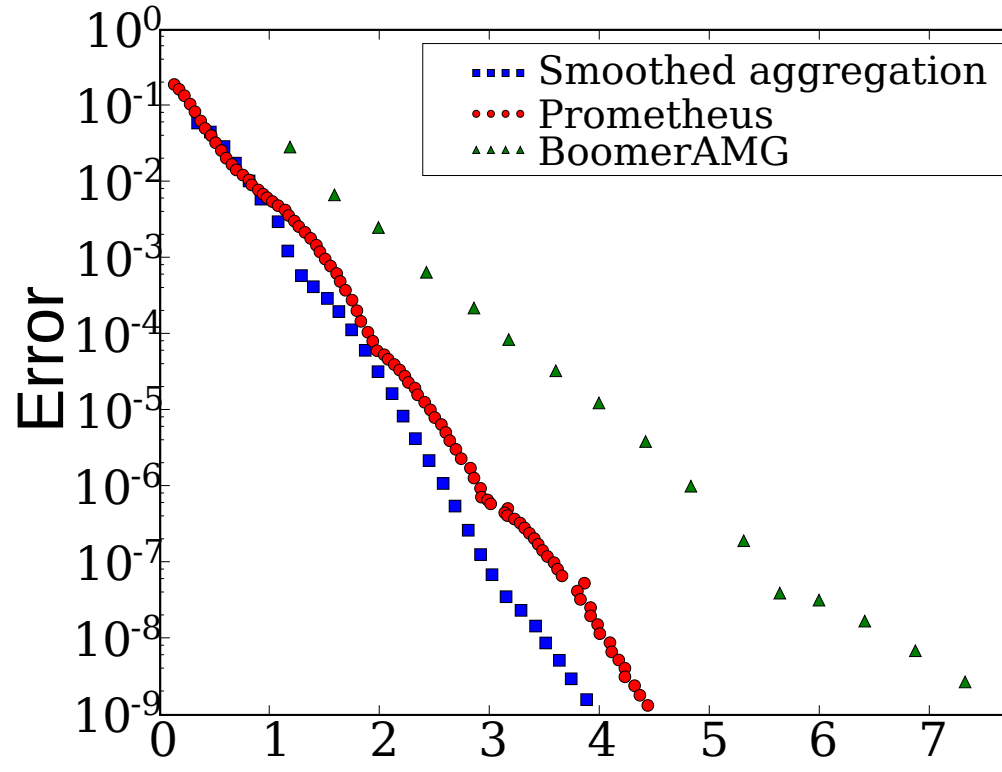
$\leftarrow$ grid 200×20

grid 2000×200:

Preconditioner	iterations	time
Smoothed agg.	6	1.42
BoomerAMG	7	2.12
Prometheus	3290	507
ICC	11.200	1677
SSOR	$\gg$ 50.000	$\gg$ 7950

OODC test case 32 km $\times$ 2 km

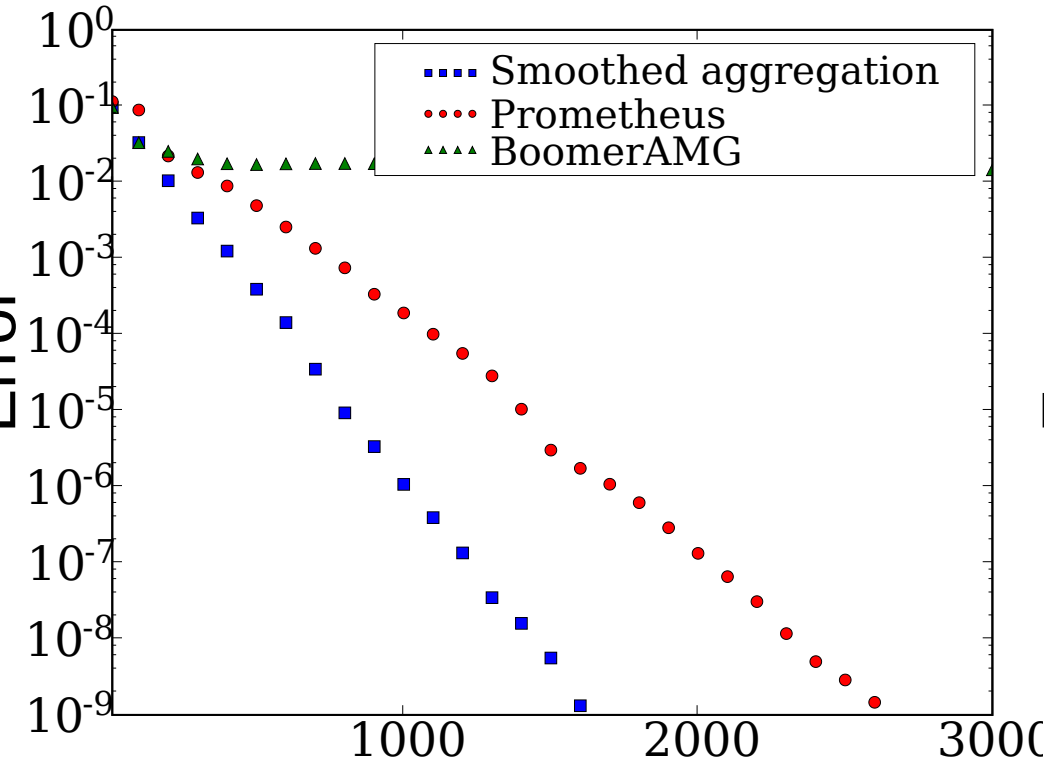


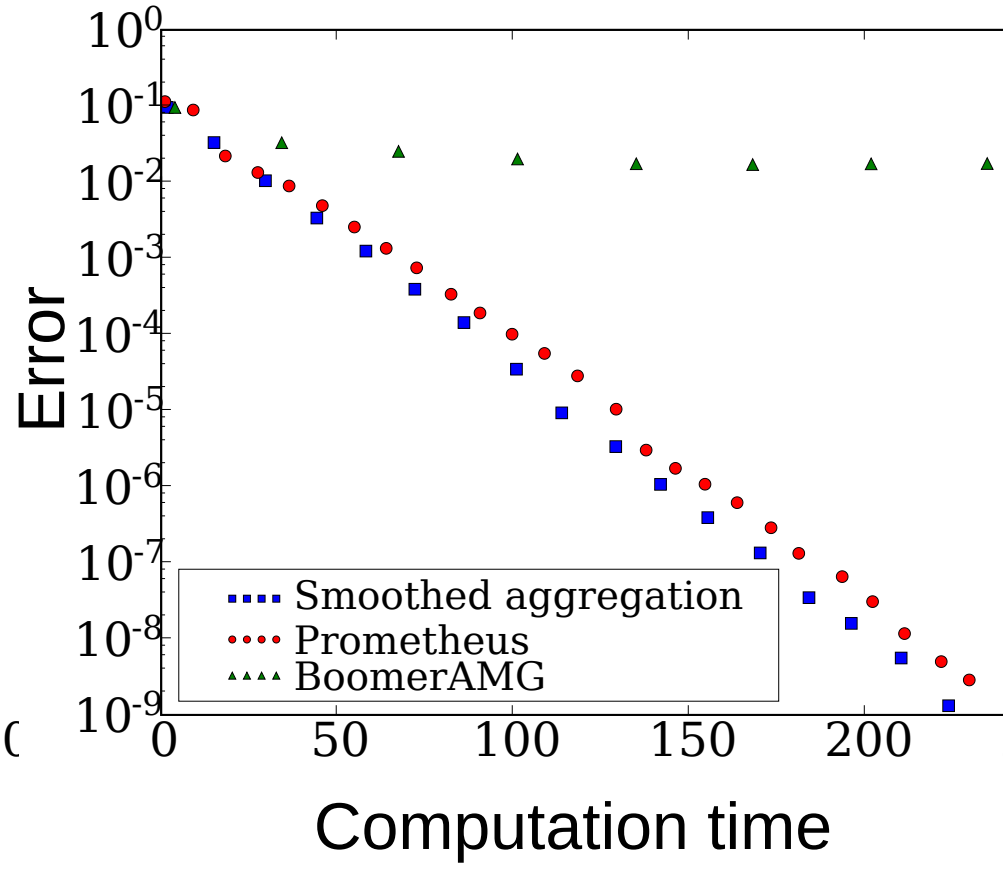


to reach 10^{-9}:

Preconditioner	iterations	time
ICC	530	17.3
SSOR	660	28.6

OODC test case 3200 km $\times$ 2 km





Traditional preconditioners ICC and SSOR not converging.

Summary

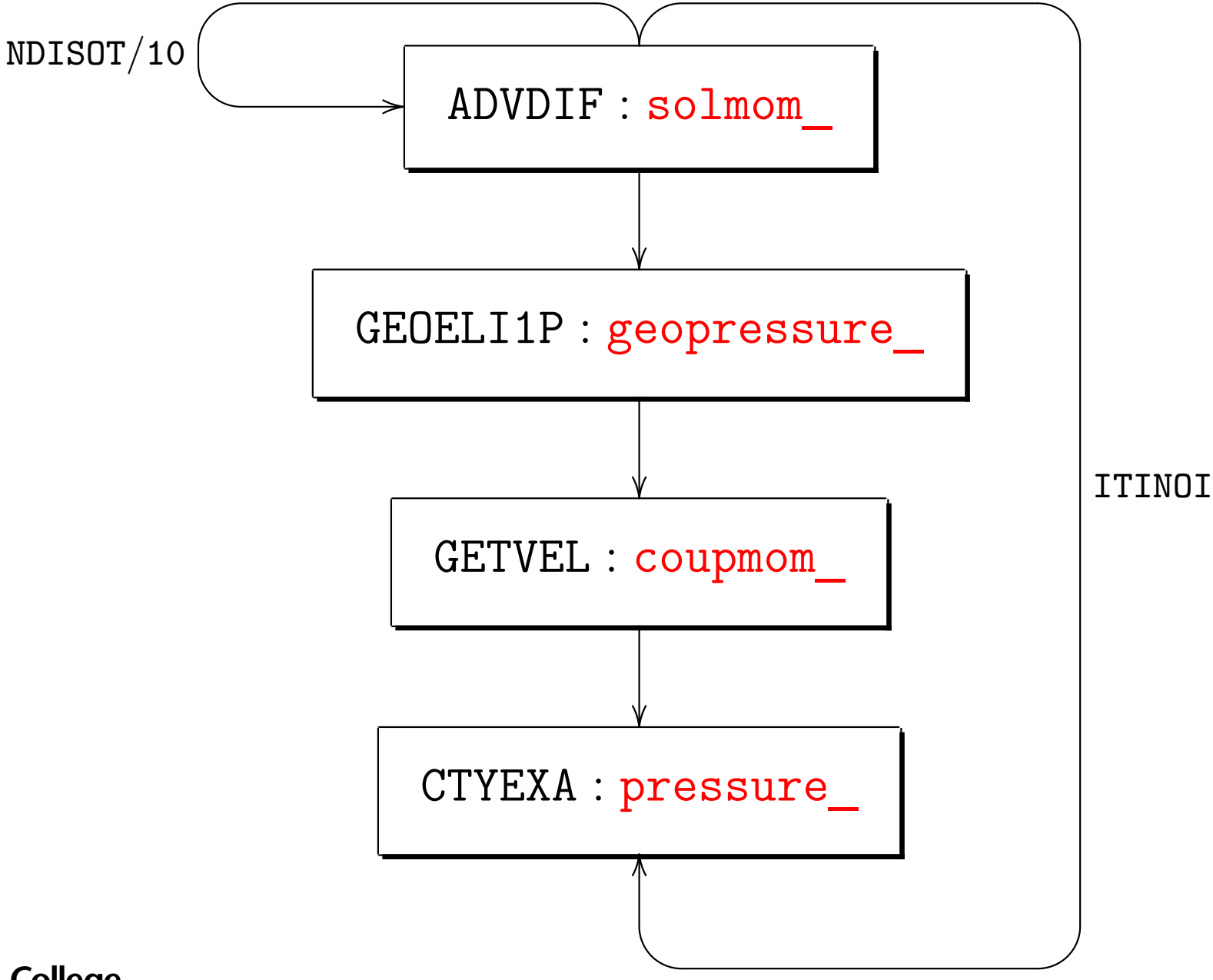
The solution of the 3D pressure Poisson problem for, in particular large aspect ratio, and more generally multi-scale problems requires advanced preconditioning techniques.

It is necessary to reduce the components of the error associated with the different length scales in seperate solves in a multi-grid approach.

This is true for layered models, where the horizontal and vertical modes are separated, but also for fully unstructured 3D models where we can apply this to the whole range of length scales that we are able to resolve.

Linear solvers in fluidity

Solves in fluidity time step



Grep for iterations!

```
[skramer@ese-luigi]{~/tst/DOME_logs} grep iteration fluidity.log-0 |  head -n25
 Start of another nonlinear iteration; ITS,ITINOI=           1           2
solmom_PETSc n/o iterations: 4
solmom_PETSc n/o iterations: 0
solmom_PETSc n/o iterations: 0
geopressure_PETSc n/o iterations: 3000
pressure_PETSc n/o iterations: 1700
coupmomentum_PETSc n/o iterations: 31
pressure_PETSc n/o iterations: 1700
 acctim,IP,iteration=   0.00000000000000                1           1
 Start of another nonlinear iteration; ITS,ITINOI=           2           2
solmom_PETSc n/o iterations: 6
solmom_PETSc n/o iterations: 5
solmom_PETSc n/o iterations: 5
geopressure_PETSc n/o iterations: 3000
coupmomentum_PETSc n/o iterations: 31
pressure_PETSc n/o iterations: 1700
 acctim,IP,iteration=   0.00000000000000                1           2
 Start of another nonlinear iteration; ITS,ITINOI=           1           2
solmom_PETSc n/o iterations: 6
solmom_PETSc n/o iterations: 5
solmom_PETSc n/o iterations: 5
geopressure_PETSc n/o iterations: 3000
coupmomentum_PETSc n/o iterations: 33
pressure_PETSc n/o iterations: 1700
 acctim,IP,iteration=   2500.00000000000                1           1
```

Closer look at the log

```
 HERE 1 just inside MULGRIDVERT
 HERE 2 just inside MULGRIDVERT
 ************PRESUSRE ITS=           1
 CORR3D: T
 Going into petsc_(block_)solve_csr.
 Solving matrix type: geopressure_
 Number of rows ==       112995
 Assembling matrix: geopressure_
 Matrix assembly completed: geopressure_
 Assembling RHS: geopressure_
 RHS assembly completed: geopressure_
 ksp_type:cg
 pc_type: sor
 ksp_max_it, ksp_atol, ksp_rtol, ksp_dtol:         3000  1.0000000000000000E-009
 1.0000000000000000E-100  -2.00000000000000
 startfromzero: T
 Entering solver: geopressure_
geopressure_PETSc reason of convergence: -3
geopressure_PETSc n/o iterations: 3000
 Out of petsc_(block_)solve_csr.
 kits,TNOIT3D1=        3000        3000
top-level-geopressure_norm:  4.1543E+02
 OUT OF MULGRIDVERT
 just out of SOLCG
 Min, max of PG =   -3.24181908742463        3.24057799102683
 just out of PGMOME
 ******PG MIN&MAX:  -3.24181908742463        3.24057799102683
```

Solver options

`-ksp_type` choice of Krylov Subspace, options: cg, gmres, fgmres, bicgs, cgs, ...

`-pc_type` choice of preconditioner, options: sor, eisenstat, ilu, cholesky, prometheus, hypre[a], mg, ...

`-ksp_max_it` maximum number of iterations

`-ksp_atol` absolute error tolerance

`-ksp_rtol` relative error tolerance

`-ksp_dtol` divergence tolerance

`-startfromzero` whether to start from zero

[a] in combination with `-pc_hypre_type boomeramg`

Option 'hierarchy'

Defaults with no PETSc options

Error bounds and maximum number of iterations from gem options: PREERR, MOMER1, MOMER2 . . . , resp. PRENOI, MOMNO1/2, TNOIT1/2, . . . or hard coded. Sensible choice of 'ksp_type' and 'pc_type' hard coded.

May be overridden by:

PETSc options that apply to all matrices

For example:

```
dfluidity -p -ksp_atol,0,-ksp_dtol,1e-6 lock_exchange
```

May be overridden by:

matrix specific PETSc options

For example:

```
dfluidity -p -geopressure_pc_type,prometheus gyre
```

Error bounds

Error bounds are based on preconditioned residual: $M^{-1}r^k$

Absolute error tolerance:

$$\|M^{-1}r^k\| \leq \epsilon, \quad \texttt{-ksp\_atol } \epsilon$$

Relative error tolerance:

$$\|M^{-1}r^k\| \leq C\|M^{-1}b\|, \quad \texttt{-ksp\_rtol } C$$

Divergence tolerance:

$$\|M^{-1}r^k\| > D\|M^{-1}b\|, \quad \texttt{-ksp\_dtol } D$$

Trouble shooting

If a linear solve fails in fluidity it will:

- Put big warnings in the log.
- Tell you why it didn't succeed, e.g.: `KSP_DIVERGED_ITS, KSP_DIVERGED_DTOL, KSP_DIVERGED_NAN`.
- Dump the matrix equation it was trying to solve in a file called `matrixdump`.
- Stop the run at the end of the time step with the usual final dump. Only if the reason of failure was `KSP_DIVERGED_ITS` (i.e. it didn't reach the required error tolerance within the maximum number of iterations) the run will continue. This is only useful for spinning up a model, the results should not be trusted as long as the solves don't converge.

Trouble shooting

What to do if a linear solve fails in fluidity:

- Check that your model results are reasonable before the first failing solve (for instance to see if it is not blowing up, or if there are NaNs).
- Check your mesh.
- Only if you are reasonably certain that is this the actual equation you want to solve try changing the solver options. For this purpose you could use `petsc_readnsolve`.